Introduction to Computer Graphics

Section 5 : http://bit.ly/1lLGqCP
Sheet 5 : http://bit.ly/1iPjhty

Younies Saeed Mahmoud
younies.mahmoud@gmail.com
https://www.facebook.com/younies.mahmoud

Sheet 4: Question 1

The following OpenGL program displays a sphere by approximating it in terms of rectangles and triangles.

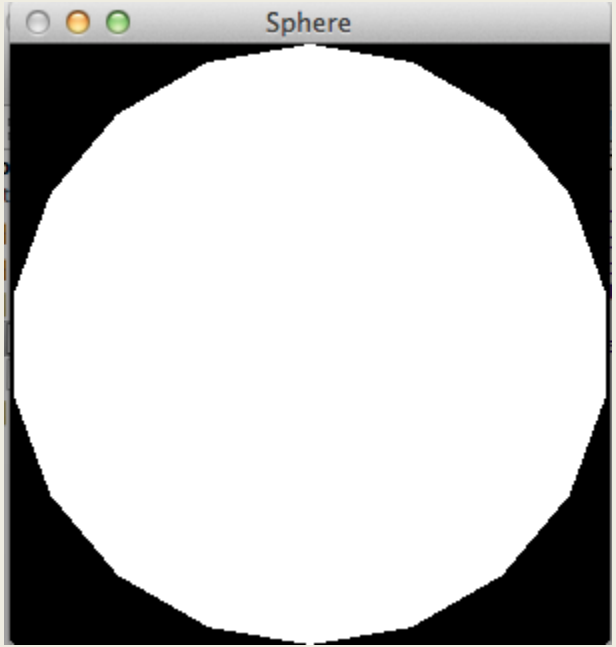
a) Write and execute the program in the link below. Does your program output appear as a sphere? If not explain why.

b) Change the step taken along both theta and phi angles to 5, 10, 25 degrees and rerun your program. What you note? Explain.

http://pastebin.com/8LFP5WZz

Answer 1:

After running



Answer1:

A. The program displays a circle instead of a sphere. The reason for this is **viewing setting**. The program **does not set** any viewing setting. Hence, the viewing is set to the default in OpenGL which makes the viewing volume as **a cube 2x2x2** centered at the origin. The contents of this cube are projected on the image plane using **orthogonal projection**. Since our sphere is centered at the origin, it appears as a circle instead of a sphere.
B. By changing the angle steps, we control the approximation. **Smaller angle steps should give smoother** approximation. The reverse is also **true**.

Question 2

Describe how you would adapt the RGB-color model in OpenGL to allow you to work with a subtractive color model (Problem 2.8).

Answer 2:

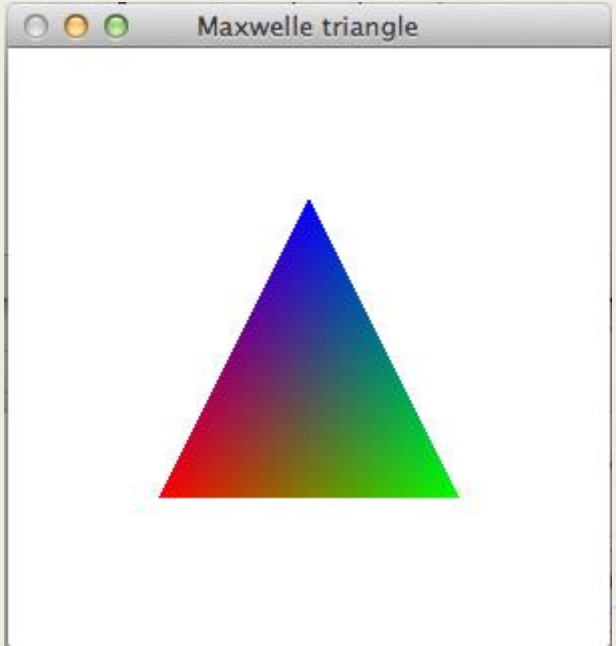
The following equations is used to convert from CMY subtractive color model to the RGB mode R=1.0-C, G=1.0-M, B=1-Y.

Question 3:

In OpenGL, we can associate a color with each vertex. If the endpoints of a line segment have different colors assigned to them, OpenGL will interpolate between the colors as it renders the line segment. It will do the same for polygons. Use this property to display the Maxwell triangle: an equilateral triangle whose vertices are red, green, and blue

Answer 3

http://pastebin.com/zGF9BiMY



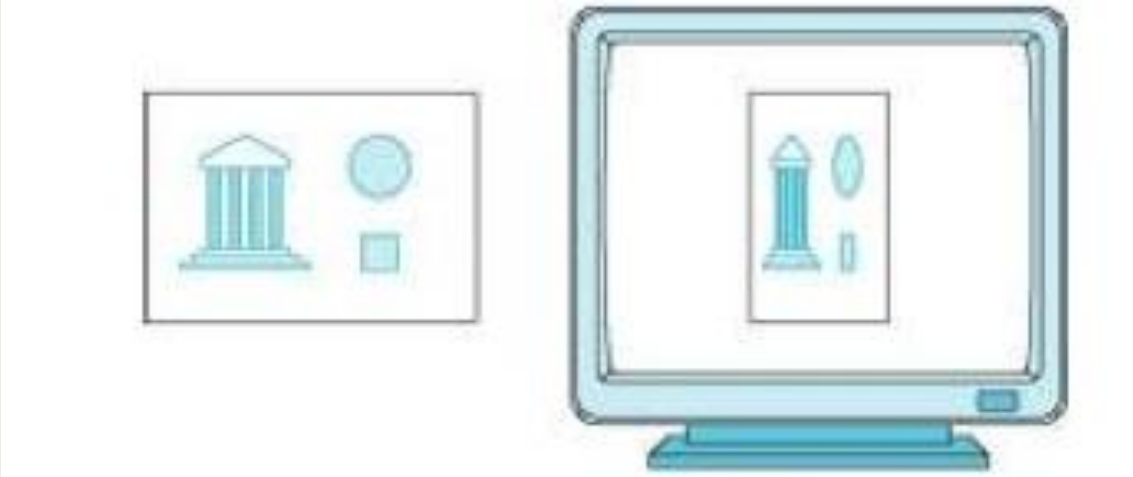
Question 4 & Question 5

REPORT

Sheet 5 _ Question 1

The following figure shows a scene that appears deformed when displayed on the output screen of an OpenGL program

A. Discuss possible reasons that could lead to the shown deformation
B. How you can avoid such deformations?



Answer 1

This deformations occurs because the aspect ratio of the **clipping window** on the **projection plan does not** have the same aspect ratio as the **display window** or **the current viewport**

To correct this problem we can do that on the following

- Using the **default viewport** (the entire window) and making the aspect ratio of the window the same as the aspect ratio of the clipping window by changing the window size or changing the size of the clipping window
- In case of drawing on a portion of the screen window using a viewport, we must choose the size of the viewport and the size of the clipping window so that they have the same aspect ratio.

Answer 1

The function used to set a view port is as follows:

```
void glViewport( GLint x, GLint y ,  GLsizei width ,  GLsizei height );
```

The function used to set the size of the window is as follows

```
glutInitWindowSize(200, 400);
```

Question 2:

Write an OpenGL program to draw a **damped** cosine functions four times, each in a separate quarter in the output graphics window. Hint; use the **viewport** setting to change the location and size of the output graphics area with respect to the output graphics window.

Answer 2

http://pastebin.com/7nT39aur

Question 3


many
Thanks!